

Fuzzy Logic (2)

Part 2 (final): Fuzzy Design

By Owen Bishop

O.and.A.Bishop@bigpond.com

There are a number of programs written for PCs and other computers that allow the user to design and debug fuzzy systems. **Figure 1** shows a screen produced by fuzzyTECH™, which is published by Inform. This is the software we used to design the dishwasher controller described in Part 1. We will look at its operation stage by stage. The rearmost window shows the Project Editor, which displays the sections of the controller. From left to right, these are the Input (Load), the Rule Block, and the Output (Washtime).

Crisp input

The input to the controller is set by a slider in a Debug window at top left. The input panel at lower left shows the input membership functions of the dishwasher as depicted in Figure 3 of Part 1. The load input has been set to 36.62 units. The input panel shows a vertical line at 36.62, which cuts the functions for 'small' and 'medium' at 0.67 and 0.33 respectively. Rules 2 and 3 are fired:

- 2 IF load is 'small',
THEN washtime is 'reduced'.
- 3 IF load is 'medium',
THEN washtime is 'medium'.

Crisp output

Output is calculated using a centre of area method and is displayed in the output window at bottom right. The 'reduced' and 'medium' output membership functions are being activated with degrees of applicability 0.67 and 0.33 respectively, as shown by the broad black arrows. The resulting crisp output value is indicated by the arrow at the bottom of the panel. As can be seen on the right of the debug panel, the crisp output is 21.655 minutes. Summing up, a load of 36.62 is to be washed for 21.655 minutes.

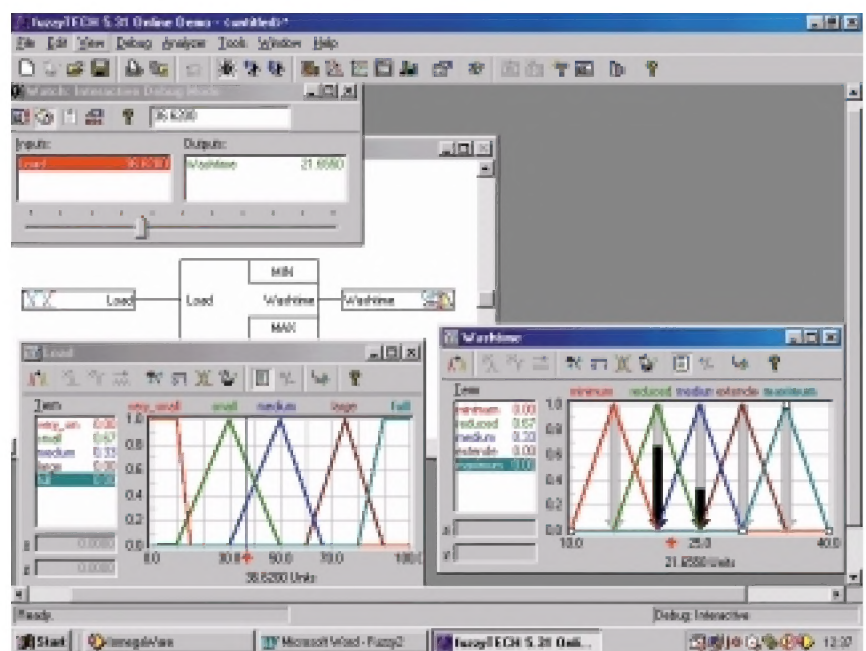


Figure 1. A fuzzy controller for a dishwasher being designed with fuzzyTECH™.

The slider on the debug panel can be set to any load value in the range 0 to 100, and the corresponding washtime is immediately displayed as the output. The displays in the input and output windows are updated at the same time. This is an extremely simple example that has only five input and output variables, and only five rules but it illustrates the essential features of fuzzy controllers.

Two inputs

To exploit the abilities of fuzzyTECH™ more fully, though by no means demonstrating all of its many and varied functions, we provide the washing machine controller

with a second input variable, dirt. This could be measured (again on an arbitrary scale) by a photometric technique that measures the cloudiness of the washing water after the machine has been running for a fixed period, such as one minute. Let this be expressed on the scale 0 to 10, and suppose that there are seven linguistic variables for dirt. These are 'almost clean', 'slight', 'low', 'medium', 'medium-high', 'high', 'very high'. The membership functions are shown in the panel at lower left in **Figure 2**.

The screendump also portrays the load input window (top right), the output window (bottom right) and

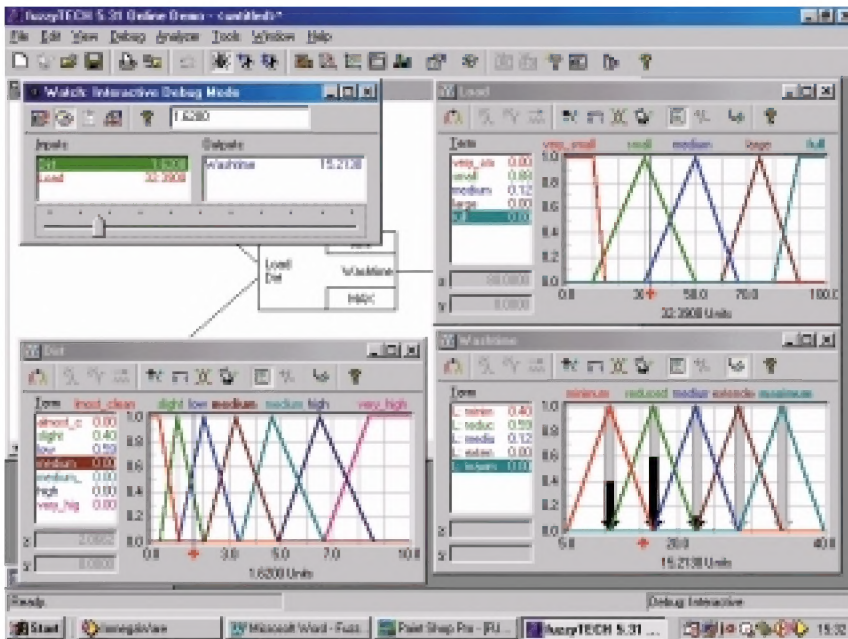


Figure 2. In this simulation, the washtime of the dishwasher depends both on the load and on the amount of dirt on the dishes.

debug window (top left).

As before, the computer generates a rule block, which contains all possible rules for connecting two input variables (at 5 and 7 levels) with one output variable at five levels. There are 175 such rules of the form:

IF <load variable>

AND <dirt variable>
THEN <washtime variable>

The user deletes unwanted rules and simplifies some of them. For example, if the dirt level is 'very high', it is advisable to wash for 'maximum time', whatever the load size. Eventually we reduced the list to 31. In general

terms, these gave increased washtime for larger loads and greater dirt. In the figure, load is set to 32.3900 and dirt to 1.6200. The load window shows that the load is in the 'small' and 'medium fuzzy sets, with indices 0.88 and 1.2 respectively. Dirt is in the 'slight' and 'low' sets with indices 0.40 and 0.59. Four of the 31 rules are fired:

- 9 IF load is 'small'
AND dirt is 'slight'
THEN washtime is 'minimum'.
- 10 IF load is 'small'
AND dirt is 'low',
THEN washtime is 'reduced'.
- 15 IF load is 'medium'
AND dirt is 'slight',
THEN washtime is 'reduced'.
- 13 IF load is 'medium'
AND dirt is 'low',
THEN washtime is 'medium'.

Three output variables are involved in the fired rules. The output panel at the bottom right of Figure 2 shows that output consists of 'minimum' (0.40), 'reduced' (0.59) and 'medium' (0.12). Defuzzification, gives a crisp output of 15.2130 minutes of washtime.

3-D Plot

The relationship between the variables can be plotted in 3-D, as in **Figure 3**. The overall response of the system can be seen as a reasonable one, washtime increasing with both load and dirt. The rule block may be set up using expert knowledge, although in this case it was a matter of the author's 'common sense'. The 3-D display allows the user to inspect the relationships visually. In this example, it is important to ensure that the response is a smooth one. Other applications might require gaps and discontinuities.

Crisp and fuzzy logic

The rather primitive dishwasher controller described above uses 31 fuzzy rules to define its behaviour. To achieve the same degree of smooth control with crisp logic rules requires possibly hundreds of rules. As an illustration of the problems, consider the behaviour of the washer at one particular load and dirt level. The rule would require crisp inputs and would provide a crisp output. A typical rule might read:

- IF load is above 40
AND dirt is above 1.5,
THEN washtime is 25.

This rule is not fired if the load is 39 and the

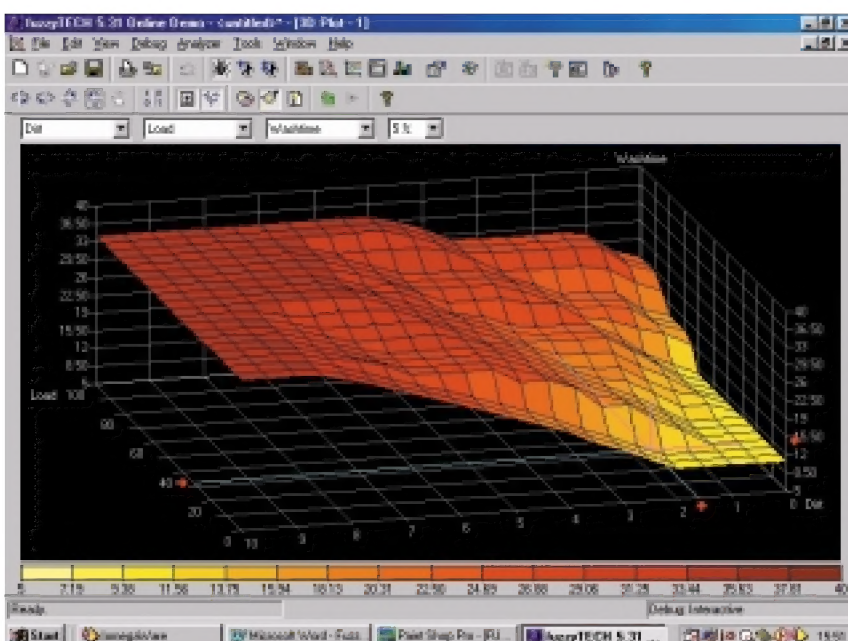


Figure 3 The response surface of the dishwasher controller shows a suitably smooth relationship between input and output.

dirt is 1.6. It is not fired if the load is 41 and the dirt is 1.49. Yet, in either case, it is obvious that the washtime should be 25, as before. Since there is no known underlying mathematical relationship that connects load, dirt and washtime, we are forced to compose reams of binary rules to cover the situation. This is the result of trying to force binary rules to a case which is not binary in its nature. In practice, most of life is fuzzy and crisp rules are inappropriate.

The advent of fuzzy logic does not mean that crisp logic has been supplanted. There are many instances in fields such as flight control, chemical processing, and boiler control where conventional control systems are satisfactorily employed. Usually these are processes where the basic mathematical relationships are known exactly or closely enough. Then we can implement the traditional techniques of proportional control (P), usually with the refinements of integral (I) and differential (D) control. However, even in these fields, fuzzy systems are beginning to challenge tradition. One of the elements in a boiler control system is the flame that heats the water. This is far from binary in nature. On the contrary, it is extremely fuzzy, so perhaps fuzzy logic could better control the boiler than the traditional systems.

It is possible to implement systems similar to the conventional P, P+I, P+D and P+I+D systems using fuzzy methods. The crisp error signal (the difference between the output and the set point of the system) can be fuzzified into, say, five fuzzy sets for use in proportional control:

'large negative', 'small negative', 'nil', 'small positive', 'large positive'.

We can similarly fuzzify the rate of change of the error signal for P+D (differential) control systems.

Applications

Fuzzy logic took a long time to gain acceptance among theoreticians and engineers of the west, especially in USA. In fact, there was considerable active opposition to the ideas. It did much better in the east, notably in Japan. There were soon a large number of practical applications that really worked, were commercially successful, and proved the advantages of the method.

Fuzzy logic is used today in a wide range of consumer goods. Washing machines using fuzzy control have been marketed by numerous companies, including Hitachi, Matsushita, Daewoo, Goldstar, Samsung, Sanyo and Sharp. These machines adjust the washing program according to the size and dirtiness of the load, the type of fabric, and the water level. They control the amount of water used, its flow rate and the wash time. At later stages in the cycle, they control rinsing and spin-drying. It is claimed that washing machines with fuzzy control complete the wash more quickly, with significant savings of electric power.

Fuzzy controllers are successfully employed in other items of domestic electrical equipment. Applications include TV sets, microwave ovens, fan heaters and air conditioners. In general, the result is superior performance coupled with power economy. An interesting application has been found in camcorders. One of the problems with these is that a hand-held camera is very much subject to camera shake. A camcorder from Matsushita eliminates this in an

ingenious way. Each frame of the recording is compared with the previous one. In normal recording some **but not all** of the parts of the image will be expected to move. If the fuzzy logic detects that all parts of the image have moved about the same distance in the same direction, this is evidence of camera shake. Accordingly, the new image is displaced to compensate for the movement and the effects of shake are considerably reduced.

Fuzzy logic has been successful on the larger scale too. A notable example is the subway system in the Japanese city of Sendai. In 1987, this was one of the first major examples of fuzzy logic control. The system controls many aspects of the subway, including the acceleration, velocity and deceleration of the trains, route control, public address systems and ticket sales. The result has been improved safety, reduced power consumption and reduced travel times.

The success of fuzzy logic is such that its use is rapidly increasing. Today's hardware, crisp on the outside, is more than likely to have a soft, fuzzy centre.

Software

At the time of writing, more detail may be had from Inform, Pascalstrasse 23, D-51000, Aachen, Germany. Their web site at <http://www.fuzzytech.com> offers a free download of the restricted version of fuzzyTECH™ that was used in the preparation of this article.

(010026-2)